1. In the below elements which of them are values or an expression? eg:- values can be integer or string and expressions will be mathematical operators.

\*

'hello'

-87.8

-

/

6

ANSWER : \*

-

/

+

-------------------------------------------------------------------------------------------------------------------------------------

2. What is the difference between string and variable?

ANSWER : Variable is a like a empty box where we can store data of a program. Variable names can are case sensitive. Eg. Variable a and A are two different variables.

Rules for Python variables:

* A variable name must start with a letter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
* Variable names are case-sensitive (age, Age and AGE are three different variables)

On other hand string is a data type. It ususally contains words or numbers or both enclosed in double or single quotes. Eg “python assign”

We can assign a string or any data type to a variable using “=”

Eg for variable and string

a = “python 1”

A = “python 2.0”

a = “python 3.1”

Here a and A are two different variables.

If I print a and A, the output will be:

2.0

3.1

Because a got updated by 3.1

-------------------------------------------------------------------------------------------------------------------------

3. Describe three different data types.

ANSWER:

There are many types of data types like string, int, float,list tuple ,range, dictionary, set,boolean

Lets see three of them

1. String - a string is a collection of one or more characters put in a single double or triple quote. It is represented as str class. Eg. A=”good ot go” , a=’1.1.1’,B = “it is 2022”.
2. Integer = it is represented by int class. It contains positive or negative whole numbers( without fraction or decimal ). In Python there is no limit to how long an integer value can be. Eg. Num = 123, num2 = 2345678987654321
3. Float - This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation. Eg. F1= 1.0, f2= 40.09

-------------------------------------------------------------------------------------------------------------------------

1. What is an expression made up of? What do all expressions do?

ANSWER:

An expression is a combination of operators and operands that is interpreted to produce some other value.where operators are special symbols that designate that some sort of computation should be performed eg +, - , / and so on. And the values that an operator acts on are called ****operands****.

Eg

A =10

B=120

A+B -5

In this case, the + operator adds the operands a and b together. An operand can be either a literal value or a variable that references an object.

A sequence of operands and operators, like a + b - 5, is called an expression. Python supports many operators for combining data objects into expressions.

1. **Constant Expressions:**These are the expressions that have constant values only.

Eg

x **=** 15 **+** 1.3

print(x)

1. **Arithmetic Expressions:**An arithmetic expression is a combination of numeric values, operators, and sometimes parenthesis. The result of this type of expression is also a numeric value. The operators used in these expressions are arithmetic operators like addition, subtraction, etc. Here are some arithmetic operators in Python

Eg

x **=** 40

y **=** 12

add **=** x **+** y

sub **=** x **-** y

pro **=** x **\*** y

div **=** x **/** y

print(add)

print(sub)

print(pro)

print(div)

1. **Integral Expressions:**These are the kind of expressions that produce only integer results after all computations and type conversions.

Eg

a **=** 13

b **=** 12.0

c **=** a **+** int(b)

print(c)

4.**Floating Expressions:**These are the kind of expressions which produce floating point numbers as result after all computations and type conversions.

Eg.

a **=** 13

b **=** 5

c **=** a **/** b

print(c)

**5 Relational Expressions:**In these types of expressions, arithmetic expressions are written on both sides of relational operator (> , < , >= , <=). Those arithmetic expressions are evaluated first, and then compared as per relational operator and produce a boolean output in the end. These expressions are also called Boolean expressions.

Eg

a **=** 21

b **=** 13

c **=** 40

d **=** 37

p **=** (a **+** b) >**=** (c **-** d)

print(p)

1. **6. Logical Expressions:**These are kinds of expressions that result in either *True*or *False.*It basically specifies one or more conditions. For example, (10 == 9) is a condition if 10 is equal to 9. As we know it is not correct, so it will return False. Studying logical expressions, we also come across some logical operators which can be seen in logical expressions most often.

P **=** (10 **==** 9)

Q **=** (7 > 5)

# Logical Expressions

R **=** P **and** Q

S **=** P **or** Q

T **=** **not** P

print(R)

print(S)

print(T)

-------------------------------------------------------------------------------------------------------------------------

1. This assignment statements, like spam = 10. What is the difference between an expression and a statement?

ANSWER :

A statement is a complete line of code that performs some action, while an expression is any section of the code that evaluates to a value. Expressions can be combined “horizontally” into larger expressions using operators, while statements can only be combined “vertically” by writing one after another, or with block constructs. Every expression can be used as a statement (whose effect is to evaluate the expression and ignore the resulting value), but most statements cannot be used as expressions.

In simple words…

a statement does something

ex: print() or spam = 10

an expression evaluates to some value

ex: 2 + 3 or 2 == 2

------------------------------------------------------------------------------------------------------------------------

6. After running the following code, what does the variable bacon contain?

bacon = 22

bacon + 1

ANSWER :

22

------------------------------------------------------------------------------------------------------------------------------------

7. What should the values of the following two terms be?

'spam' + 'spamspam'

'spam' \* 3

ANSWER:

For both of them the output will be

spamspamspam

------------------------------------------------------------------------------------------------------------------------------------------

1. Why is eggs a valid variable name while 100 is invalid?

* ANSWER:  
  There are some rules to follow to define a variable like:  
  A variable name must start with a letter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
* Variable names are case-sensitive (age, Age and AGE are three different variables)

Hence eggs is valid variable because it started with a letter and not with a number. And 100 is a invalid variable because it started with a number.

-------------------------------------------------------------------------------------------------------------------------

1. What three functions can be used to get the integer, floating-point number, or string version of a value?

ANSWER:  
We can use int() function to get Integer version value, float() function to get floating-point number, and str() to get string version value

--------------------------------------------------------------------------------------------------------------------------

10. Why does this expression cause an error? How can you fix it?

'I have eaten ' + 99 + ' burritos.'

ANSWER :

This expression can cause the TypeError like:

TypeError: can only concatenate str (not "int") to str

Because while concatenating in an expression we can concatenate only string (which is in the single , double or triple quotes).

To fix it, we can either go with:

num = "99"

'I have eaten ' + num + ' burritos.'

Or

'I have eaten ' + "99" + ' burritos.'

Or using format()

A=”I have eaten ”

B=” burritos”

Num = 99

print(“{} {} {}”.format(A,Num,B))

Or using f-string()

A=”I have eaten ”

B=” burritos”

Num = 99

print(f'{A}{Num}{B}')

OR using % operator

print("%s%s%s" % (A, Num,B))